INVESTR

Security Intensive Documentation

Team LAST

May 19, 2017

Team Members

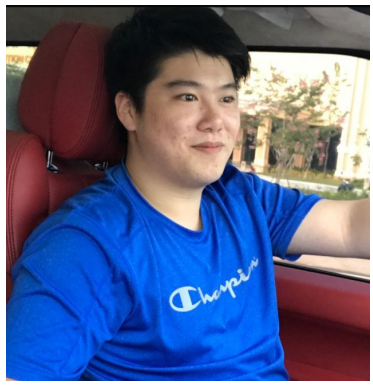
Lindsay Chung
Anthony Oeum
Sean Lin (Yu Yin Lin)

Contributors:

Team Member	Email	Role
Lindsay Chung	s3487579@student.rmit.edu.au	Product Owner
Anthony Oeum	s3484960@student.rmit.edu.au	SCRUM Master
Sean Lin (Yu Yin Lin)	s3486048@student.rmit.edu.au	Project Member

‘

Lindsay Chung (Product Owner)



Anthony Oeum (SCRUM Master)

Sean Lin (Project Member)

Top 10 Security Threats

1. **Injection** - can occur when untrusted data is sent to another user which is part of a command or query. The attacker's data which is being compromised can trick the interpreter into executing unintended commands or accessing data without proper authorization

2. **Broken Authentication and Session Management** - Application functions that are related to authentication and session management are usually implemented incorrectly which can allow attackers to get a hold of passwords, keys, session tokens and they can even get a hold of the another user's account and get assume their identity.

3. **Cross-Site Scripting** - When an application has untrusted data in a new webpage without any proper validation or updates the existing web page with user supplied data using a browser API which can create JavaScript, then XSS flaws occur. Attackers are able to execute scripts into the victim's browser, which can hijack user sessions, deface websites, or redirect users to malicious websites.

4. **Broken Access Control** - Restrictions are not properly enforced for authenticated users. This gives attackers access to unauthorized functionality and data. They are able to access other users' accounts, view sensitive files, modify user's data and change access rights.

5. **Security Misconfiguration** - A good security must have a secure configuration defined and deployed for the application, frameworks, application server, web server, database server and platform. Secure settings should be defined, implemented and controlled, since the default settings are usually insecure. Software should also be kept up to date.

6. **Sensitive Data Exposure** - Many web applications and APIs do not properly protect their sensitive data: financial and healthcare data. Attackers can potentially steal or modify weakly protected data to engage in credit card fraud, identity theft and other crimes. Sensitive data requires extra protection like encryption at rest/in transit.

7. **Insufficient Attack Protection** - Most applications/APIs lack the ability to detect, prevent and respond to both manual and automated attacks. Attack protection goes way beyond input validation and involves automatically detecting, logging, responding and blocking exploit attempts. Application owners also need to be able to release patches quickly to protect against attacks.

8. **Cross-Site Request Forgery** - A CSRF attack forces a logged-on victim's browser to send a forged HTTP request, which includes the victim's session cookie and other automatically included authentication information, to a vulnerable web application. An attack that allows the attacker to force a victim's browser to generate requests the vulnerable application thinks are the legitimate requests from the victim.

9. **Using Components with Known Vulnerabilities** - Components such as libraries, frameworks and other software modules run with the same privileges as the application. If a vulnerable component is exploited, an attack can cause major data loss or server takeover. Applications and APIs which use components that have known vulnerabilities that may undermine application defenses and enable various attacks and impacts.

10. **Underprotected APIs** - Modern applications these days usually involve rich client applications/APIs of some kind such as SOAP/XML or REST/JSON. These APIs are usually unprotected and can contain various vulnerabilities.